Amendments to the Claims

1. (Currently amended) A method for identifying to a user, the differences between elements of two hierarchically structured files, comprising the steps of:

comparing the elements of a base file to the elements of a modified file;

~~providing~~ displaying to the user a tree structure, said tree structure combining the elements of said base and said modified files; and

highlighting, in the tree structure, the differences between said elements of said base and said modified files.

2. (Original) The method of claim 1 further comprising the step of allowing the user to resolve said differences between elements, thereby creating a merged file containing elements from said base file and elements from said modified file.

3. (Previously presented) The method of claim 2 which includes indicating to the user, in the tree structure, differences between elements by one of the identifiers: new, changed or removed.

4. (Original) The method of claim 3 which includes, for an element identified as new, providing the user with the following options:

a) do not use the new element, whereby the new element is not incorporated into said merged file; and

b) use the new element, whereby the new element and children thereof, if any, are incorporated into said merged file.

5. (Original) The method of claim 3 which includes, for an element identified as changed, providing the user with the following options:

a) use old, where conflict, whereby for the merged file the changed element is taken from the base file together with unresolved children thereof, if any; and

b) use new, where conflict, whereby for the merged file the changed element is taken from the modified file together with unresolved children thereof, if any.

6. (Original) The method of claims 3, which includes, for an element identified as

removed, providing the user with the following options:

a) do not delete, whereby the merged file has the element as it exists in the base file; and

b) delete from the base file, whereby the merged file does not have the element that was deleted from the base file.

7. (Cancelled)

8. (Currently Amended) The method of claim 1, ~~7~~ wherein ~~visually~~ displaying the tree structure comprises displaying to the user a screen containing three panes, the first pane displaying said tree structure, the second pane displaying an element of said base file, and the third pane displaying an element of the modified file.

9. (Original) The method of claim 8 which includes, when the user selects an element of the tree structure displayed in the first pane, displaying the source code for the selected element:

a) in the second pane if the selected element exists in the base file; and

b) in the third pane if the selected element exists in the modified file.

10. (Original) The method of claim 1 wherein the step of comparing uses an ID attribute of the elements of the base file and the modified file being compared.

11. (Original) The method of claim 1 wherein the step of comparing uses a name attribute of the elements of the base file and the modified file being compared.

12. (Original) The method of claim 1 wherein said hierarchically structured files are XML (eXtensible markup language) files and wherein the step of comparing uses:

if provided by the elements of the base and modified files being compared, an attribute of type ID;

if an attribute of type ID is not provided by the elements of the base and modified files being compared, a <Uuid> tag if provided by the elements of the base and modified files being compared;

if an attribute of type ID and a <Uuid> tag is not provided by the elements of the base and modified files being compared, a name attribute if provided by the elements of the base and modified files being compared; and

if an attribute of type ID, a <Uuid> tag and a name attribute is not provided by the elements of the base and modified files being compared, a concatenation of a tag of the element and a value of the element.

13. (Original) The method of claim 1 wherein said hierarchically structured files are XML (eXtensible Markup Language) files.

14. (Cancelled)

15. (Original) A program storage device readable by a data processing system, tangibly embodying a program of instructions, executable by said data processing system to perform the method steps of claim 1.

16. (Currently amended) A system for identifying to a user, the differences between elements of two hierarchically structured files, comprising:

means for comparing the elements of a base file to the elements of a modified file;

means for ~~providing~~ displaying to the user a tree structure, said tree structure combining the elements of said base and said modified files; and

means for highlighting, in the tree structure, the differences between said elements of said base and said modified files.

17. (Original) The system of claim 16 further comprising means for allowing the user to resolve said differences between elements, thereby creating a merged file containing elements from said base file and elements from said modified file.

18. (Previously presented) The system of claim 17 which includes means for indicating to the user, in the tree structure, differences between elements by one of the identifiers: new, changed or removed.

19. (Original) The system of claim 18 which includes, for an element identified as new, providing the user with the following options:

a) do not use the new element, whereby the new element is not incorporated into said merged file; and

b) use the new element, whereby the new element and children thereof, if any, are incorporated into said merged file.

20. (Original) The system of claim 18 which includes, for an element identified as

changed, means for providing the user with the following options:

a) use old, where conflict, whereby for the merged file the changed element is taken from the base file together with unresolved children thereof, if any; and

b) use new, where conflict, whereby for the merged file the changed element is taken from the modified file together with unresolved children thereof, if any.

21. (Original) The system of claim 18, which includes, for an element identified as removed, means for providing the user with the following options:

a) do not delete, whereby the merged file has the element as it exists in the base file; and

b) delete from the base file, whereby the merged file does not have the element that was deleted from the base file.

22. (Cancelled)

23. (Currently Amended) The system of claim 16,~~22~~ wherein the means for ~~visually~~ displaying the tree structure comprises means for displaying to the user a screen containing three panes, the first pane displaying said tree structure, the second pane displaying an element of said base file, and the third pane displaying an element of the modified file.

24. (Original) The system of claim 23 which includes, when the user selects an element of the tree structure displayed in the first pane, means for displaying the source code for the selected element:

a) in the second pane if the selected element exists in the base file; and

b) in the third pane if the selected element exists in the modified file.

25. (Original) The system of claim 16 wherein the means for comparing uses an ID attribute of the elements of the base file and the modified file being compared.

26. (Original) The system of claim 16 wherein the means for comparing uses a name attribute of the elements of the base file and the modified file being compared.

27. (Original) The system of claim 16 wherein said hierarchically structured files are XML (eXtensible markup language) files and wherein the means for comparing uses:

if provided by the elements of the base and modified files being compared, an attribute of type ID;

if an attribute of type ID is not provided by the elements of the base and modified files being compared, a <Uuid> tag if provided by the elements of the base and modified files being compared;

if an attribute of type ID and a <Uuid> tag is not provided by the elements of the base and modified files being compared, a name attribute if provided by the elements of the base and modified files being compared; and

if an attribute of type ID, a <Uuid> tag and a name attribute is not provided by the elements of the base and modified files being compared, a concatenation of a tag of the element and a value of the element.

28. (Original) The system of claim 16 wherein said hierarchically structured files are XML (eXtensible Markup Language) files.

29. (Previously presented) A system for determining the differences between two hierarchically structured files comprising:

a parser to parse the files and produce a parse tree output for each file; and

a comparison module to compare the parse tree outputs from the parser and to create a merged tree from the parse tree outputs.

30. (Original) The system of claim 29, further comprising a tree view module to display the merged tree.

31. (Previously presented) A hierarchical data structure for use by a computer system and stored on a computer-readable storage medium, said structure comprising:

a plurality of nodes;

each of said nodes corresponding to a hierarchical element contained within a base file or a modified file, said files stored within said computer system; and

each of said nodes having an indicator indicating if said node is new, changed or removed when comparing the nodes of said base file to said modified file.

32. (New) The method of claim 1, wherein:

the step of comparing comprises comparing information associated with at least one of the elements of the base file with information associated with at least one of the elements of the modified file;

the information associated with the at least one of the elements of the base file identifies attributes of the at least one of the elements of the base file; and

the information associated with the at least one of the elements of the modified file identifies attributes of the at least one of the elements of the modified file.

33. (New) The method of claim 32, wherein:

the information associated with the at least one of the elements of the base file further identifies attributes of descendant elements thereof, if any;

the information associated with the at least one of the elements of the modified file further identifies attributes of descendant elements thereof, if any.

34. (New) The method of claim 32, wherein:

the information associated with the at least one of the elements of the base file comprises information encoded by a digest function; and

the information associated with the at least one of the elements of the modified file comprises information encoded by the digest function.

35. (New) The method of claim 1, further comprising:

identifying at least one of the elements of the base file and the modified file according to a user-customizable identity function.

36. (New) The system of claim 16, wherein:

the means for comparing comprises means for comparing information associated with at least one of the elements of the base file with information associated with at least one of the elements of the modified file;

the information associated with the at least one of the elements of the base file identifies attributes of the at least one of the elements of the base file; and

the information associated with the at least one of the elements of the modified file identifies attributes of the at least one of the elements of the modified file.

37. (New) The system of claim 36, wherein:

the information associated with the at least one of the elements of the base file further identifies attributes of descendant elements thereof, if any;

the information associated with the at least one of the elements of the modified file

further identifies attributes of descendant elements thereof, if any.

38. (New) The system of claim 36, wherein:

the information associated with the at least one of the elements of the base file comprises information encoded by a digest function; and

the information associated with the at least one of the elements of the modified file comprises information encoded by the digest function.

39. (New) The system of claim 16, further comprising:

means for identifying at least one of the elements of the base file and the modified file according to a user-customizable identity function.